IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the application of: CLICK et al.)	
)	Group Art Unit: 2192
)	
)	Examiner: Kendall, Chuck O.
Application No: 09/872,458)	
)	Atty. Docket No.: SUNMP018
)	
Filed: May 31, 2001)	
)	Date: July 5, 2006
For: System and Method for Range Check)	
Elimination Via Iteration Splitting in a)	
Dynamic Compiler)	

PRE-APPEAL BRIEF

Mail Stop: AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Final Office Action of April 6, 2006, Applicant respectfully requests a pre-appeal review of the outstanding rejections in accordance with the Pre-Appeal Brief Conference Pilot Program announced on July 12, 2005 (1296 OG 67) which was extended on February 7, 2006 (1303 OG 21). In accordance with these procedures, Applicant respectfully submits that the present Application should be allowed for the reasons stated in the Remarks, which begin on the following page.

REMARKS

This Pre-Appeal Brief is submitted in response to the Final Office Action of April 6, 2006 (hereinafter "the Office Action"). All references herein to the claims, except as noted, will be made with reference to the claim list provided in the Amendment submitted previously on January 17, 2006. References to line numbers in the Office Action, except as noted, will count every printed line, except the page header, but including section headings. If there is any confusion or questions regarding any aspect of this Pre-Appeal Brief, the Examiner and other officials reviewing this Pre-Appeal Brief are invited to contact the undersigned.

For the purposes of this Pre-Appeal Brief, only the primary arguments directed to the independent claims are presented herein. Additional arguments, e.g., directed to the subject matter of the dependent claims, may be presented if and when the case proceeds to Appeal.
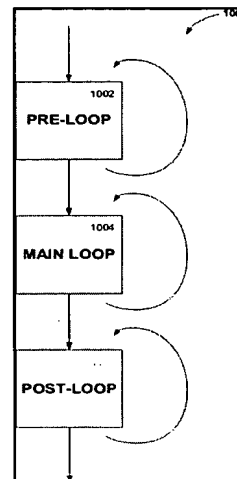
Status

All pending claims (claims 1-6 and 27-38) stand rejected under 35 U.S.C. § 103(a) for being unpatentable over U.S. Patent 6,519,765 to Kawahito et al. (hereinafter, "Kawahito") in view of U.S. Patent 5,797,013 to Mahadevan et al. (hereinafter, "Mahadevan").

Independent Claims

Independent claims 1, 27, 31, and 35 relate to methods for speeding up the execution of compiled computer programs by reducing the number of array boundary checks. As is generally understood in the field, variable arrays, which generally contain number of related values, are accessed using an index expression which identifies one of the values in the array. The index expression is used by the compiler to identify the address in memory of the particular value being referenced. Because variable arrays are assigned only a finite amount of memory, if the index expression is too great or too small, referred to in the art as "overflow" and "underflow," respectively, it is possible that data outside the area of memory assigned to the array could be accessed, which could cause stability problems in executing the program. Therefore, compilers often insert range checks to ensure that the index variable falls within the limited range of values corresponding to assigned memory. These range checks are typically made each time the array is accessed.

It is common for programmers to employ an execution loop when reading or setting a range of values of an array by incrementing the index variable. In these cases, the index variable is used to determine when to exit the loop. When an index variable is used to test whether to exit a loop, it is referred to as a trip counter. The presently claimed embodiments take advantage of this by taking a single execution loop in the source code and converting it into three loops in the compiled code, the three loops being executed in series. Such a conversion is referred to in the art as "iteration splitting." Depending on the index expressions used within the loop, certain iterations of the original loop can be placed in a main loop portion that has no range checking, other iterations placed in a pre-loop or post-loop which performs only range checks for underflow or overflow conditions, respectively.

Application, Figure 10

In addition to iteration splitting described above, each of the independent claims currently set forth the operation of sorting index expressions by the trip counter and offset portions of the index expressions for each of the arrays accessed using the index expression. The trip counter is a variable that identifies when a loop should be exited, and the offset is any amount added to or subtracted from the trip counter. See, e.g., Figure 8 and associated text in the second and third full paragraphs of page 18 of the present Application. The sorted expressions are used to determine the resulting loop structures as described in the Application with reference to Figure 9. None of the references teach these operations.

1.   The prior art fails to teach or suggest all claim limitations

Kawahito is cited in the Office Action as showing all the limitations of claim 1 except sorting the index expressions by the trip counter and offset (Office Action, page 3, line 13). However claim 1 sets forth, "creating a loop structure using iteration splitting . . . the loop structure being determined based on the sorted index expressions" (claim 1, lines 7-11). The Office Action appears to suggest that Kawahito teaches this concept at column 2, lines 50-60 (Office Action, page 3, lines 1-5). This portion of Kawahito relates to a prior art method of eliminating array range checks by using information in an *if* statement as shown in Table 5 of Kawahito. The Office Action admits that Kawahito does not mention sorting index expressions (page 3 lines 12-14). Therefore, it is not possible for Kawahito to teach creating a loop structure using iteration splitting where the loop structure is determined based on the sorted index expressions.

The Office Action cites Mahadevan for teaching sorting index expressions (Office Action, page 3, lines 14-16). However, Mahadevan does not mention sorting index expressions by both trip counter and offset as claimed. See, for example, claim 1, line 6; claim 27, line 6; claim 31 line 11, and claim 35 line 6. Since Mahadevan does not teach sorting index expressions by both index and offset as set forth in the claims, this limitation is lacking in the prior art. Furthermore, the limitation of "creating a loop structure using iteration splitting . . . the loop structure being determined based on the sorted index expressions" is also not met since Mahadevan does not mention iteration splitting or any analogue thereof. In particular, Mahadevan teaches sorting the index expressions only to determine maximum distance between them, to calculate an optimum loop unrolling factor (col. 9, lines 36-38). Thus, the prior art does not teach creating a loop structure using iteration splitting wherein the loop structure is determined based on the sorted index expressions as essentially set forth in the independent claims.

Since the claim features discussed above are not taught or suggested by the prior art, Applicant respectfully submits the rejection under 35 U.S.C. § 103(a) is improper and should be withdrawn as it pertains to claim 1. Since claims 2-6 depend from claim 1, they are allowable for at least the same reasons as claim 1. Independent claims 27, 31, and 35 contain similar limitations to that described above with respect to claim 1 (cliam 27, lines 7-9; claim 31, lines 12-14; and claim 35, lines 6-9). Therefore, claims 27, 31, and 35 as well as claims depending therefrom, should be found patentable over the prior art of record.

2.   The prior art fails to provide motivation to combine Kawahito and Mahadevan.

The Office Action asserts that, "it would have been obvious . . . to combine Kawahito and Mahadevan because, sorting the index expression would enable determining the maximum distance between them and hence make the program more efficient during optimization" (page 3, lines 20-24). Applicants respectfully disagree. In particular, Applicants see no basis for the suggestion that knowing the maximum distance between index expressions are useful in making the program more efficient during optimization, other than in the context of loop unrolling, which has nothing to do with reducing array boundary checks as described by Kawahito and as claimed herein.

In fact, as mentioned in the Office Action, Mahadevan teaches listing the various indexes in the loops and sorting them to "notice the maximum distance between them" (Mahadevan, col. 9, line 34-35; Office Action page 3, lines 18-20). Mahadevan only teaches

that it is helpful to know the distance between index expressions to determine a loop unrolling factor. As explained in Mahadevan, the loop unrolling factor may be calculated by adding one to the maximum difference between two index expressions (col. 9, lines 36-37). The unroll factor is the number of times a loop body is repeated in an unrolled loop. Thus, if a particular loop is executed N times, and the unroll factor is four, then the loop body is repeated four times and this unrolled loop is executed N/4 times (col. 2, line 64 to col. 3, line 8).

Since the technique of Kawahito (iteration splitting) and that of Mahadevan (loop unrolling) are two separate techniques, and because the maximum distance is determined in Mahadevan to determine an optimum loop factor – a concept not relevant or useful to Kawahito – Applicants respectfully submit that there was no motivation in the prior art for a person wishing to reduce the number of array boundary checks in compiled code to sort the index expressions.

Conclusion

In view of the foregoing, the Applicant respectfully submits that all of the pending claims are in condition for allowance. The Applicant kindly requests that the Office withdraw the rejections of claims 1-21, and issue a Notice of Allowance. If the Office has any questions concerning the present Pre-Appeal Brief, the undersigned can be reached at (408) 749-6900 Ext. 6933. If any additional fees are due in connection with filing this Pre-Appeal Brief, the Commissioner is authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP018). Enclosed herewith is the associated Notice of Appeal and Return Receipt Postcard.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP

Leonard Heyman
Reg. No. 40,418

Martine Penilla & Gencarella, LLP
710 Lakeway Drive, Suite 200
Sunnyvale, California 94086
Tel: (408) 749-6900 Ext. 6933
Customer Number 32291